

AMENDMENTS TO THE CLAIMS

1. (Currently amended) A computerized method for collaborating over a network to manipulate a design representing electrical or mechanical assemblies using a plurality of heterogeneous user applications running on respective clients connected to the network, said method comprising the steps of:

connecting a session client process to a session manager over the network to participate in a collaborative session;

sharing session control messages with other session client processes connected to said session manager;

loading design data representing said design into a local application running on said client;

creating at least one application state file representing at least one application state of said local application based on at least one manipulation of said design using said local application;

communicating said at least one application state file from said session client process to said other session client processes via said session manager; and

receiving loading at least one application state file created by other local applications and communicated from said other session clients via said session manager;

presenting the at least one application state file created by other local applications to a user;

allowing the user to delay the instantiation of the at least one application state file created by other local applications; and

loading the at least one application state file created by other local applications, thereby allowing the user to manipulate a first aspect of the design before loading changes made to a second aspect of the design by another user.

2. (Original) The method of claim 1 wherein said at least one application state is encoded using normalized XML structures to create said at least one application state file, and wherein said at least one application state file is communicated as an XML message.

3. (Original) The method of claim 2 wherein said XML structures are based on domain specific conventions defined in the context of the type of design data.
4. (Original) The method of claim 1 further comprising saving said session controls and said at least one application state file in a journal file.
5. (Original) The method of claim 1 further comprising the step of scheduling said collaborative session.
6. (Original) The method of claim 1 further comprising the step of conducting a text-based conversation with said other session clients.
7. (Original) The method of claim 1 further comprising the steps of logging in to said collaborative session and logging out of said collaborative session.
8. (Original) The method of claim 1 further comprising the step of controlling the loading of said application state file at a time selected by the user.
9. (Original) The method of claim 1 further comprising the step of displaying design manipulations corresponding to said application state file created and communicated by said other application files.
10. (Original) The method of claim 1 wherein said design is manipulated without having to transmit design images between said heterogeneous applications.
11. (Currently amended) A computerized method for ~~collaborating~~ providing asynchronous training to a user to manipulate a design representing electrical or mechanical assemblies using a plurality of heterogeneous applications, said method comprising the steps of:
 - loading design data into a local application on a computer;
 - manipulating a design based on said design data using said local application;

creating at least one application state file representing an application state of said local application based on at least one manipulation of said design using said local application;

saving each said at least one application state file in a journal file; and

transmitting said journal file to another computer such that said at least one application state file can be loaded on said another computer and said at least one manipulation can be reviewed using another heterogeneous application running on said another computer, wherein said journal file provides interactive instructions when played back on said another computer, thereby providing asynchronous training to the user.

12. (Original) The computerized method of claim 11 wherein said method is an asynchronous method of collaboration.

13. (Cancelled)

14. (Previously presented) A computerized method for dynamically collaborating between heterogeneous applications running concurrently on a single client, said method comprising the steps of:

loading design data representing electrical or mechanical assemblies into one of said applications on said client;

manipulating a design object using said one of said applications;

creating at least one application state file representing at least one application state of said design data based on at least one manipulation of said design object using said one of said applications;

dynamically notifying at least one other of said applications of said application state file using local operating system inter-process messaging;

reading said application state file using said at least one other of said applications; and

manipulating a corresponding design object in said at least one other of said applications based on said application state file.

15. (Original) The computerized method of claim 14 wherein the step of manipulating said design includes highlighting said design object, and wherein said other of said applications highlights said corresponding design object upon reading said application state file.

16. (Original) The computerized method of claim 14 wherein said heterogeneous applications collaborate bi-directionally.

17. (Currently amended) A computer program product comprising a storage medium having computer-readable code stored thereon for collaborating over a network to manipulate a design representing electrical or mechanical assemblies using a plurality of heterogeneous applications running on clients connected to the network, said system comprising:

- at least one application state file for representing application states created by a local application based on at least one manipulation of a design made using said local application; and

- a session client module for synchronizing interaction between said local application and a collaborative session and for communicating said application state file over the network to other clients and receiving at least one other application state file from the other clients, presenting the at least one other application state file to a user, allowing the user to delay the instantiation of the at least one other application state file, and loading the at least one other application state file, thereby allowing the user to manipulate a first aspect of the design before loading changes made to a second aspect of the design by another user.

18. (Previously presented) The computer program product of claim 17 wherein said session client module comprises:

- a user interface for interfacing with a user;
- a first application programming interface for interfacing with said local application; and
- a second application programming interface for interfacing with a session manager over the network.

19. (Previously presented) The computer program product of claim 17 further comprising a session journal for recording session controls and said application states during said collaboration session.

20. (Previously presented) The computer program product of claim 17 wherein said application states are encoded using normalized XML structures to create said at least one application state file, and wherein said at least one application state file is communicated as an XML message.

21. (Previously presented) The computer program product of claim 17 wherein said session client module communicates said application state file without sending images of said design.

22. (Currently amended) A computer program product comprising a storage medium having computer-readable code stored thereon for ~~collaborating~~ providing asynchronous training to a user to manipulate a design representing electrical or mechanical assemblies using a plurality of heterogeneous applications, said system comprising:

means for loading design data into a local application on a computer;

means for manipulating a design based on said design data using said local application;

creating at least one application state file representing an application state of said local application based on at least one manipulation of said design using said local application;

means for saving each said at least one application state file in a journal file; and

means for transmitting said journal file to another computer such that said at least one application state file can be loaded on said another computer and said at least one manipulation can be reviewed using another heterogeneous application running on said another computer, wherein said journal file provides interactive instructions when played back on said another computer, thereby providing asynchronous training to the user.

23. (Currently amended) A computer program product comprising a storage medium having computer-readable code stored thereon for collaborating over a network to manipulate a design representing electrical or mechanical assemblies using a plurality of heterogeneous applications on client computers connected to the network, said computer-readable code comprising:

code for interfacing with a user;

code for interfacing with a local application used to manipulate a design;

code for interfacing with a session manager over a network; ~~and~~

code for creating a session client process to communicate session controls and at least one application state file to said session manager over said network, wherein said application state file is created by said local application based on at least one manipulation of said design; and

code for receiving at least one other application state file from said session manager over said network, presenting the at least one other application state file to the user, allowing the user to delay the instantiation of the at least one other application state file, and loading the at least one other application state file, thereby allowing the user to manipulate a first aspect of the design before loading changes made to a second aspect of the design by another user.

24. (Currently amended) A computer program product comprising a storage medium having computer-readable code stored thereon for collaborating over a network to manipulate a design representing electrical or mechanical assemblies using a plurality of heterogeneous applications on client computers connected to the network, said computer-readable code comprising:

code for interfacing with a user;

code for interfacing with a local application;

code for instructing said local application to create at least one application state file representing at least one application state based on a manipulation made to a design using said application state file;

code for interfacing with a session manager over a network;

code for sending said at least one application state file and session controls to said session manager; and

code for notifying said local application that an at least one other application state file has been received, presenting the at least one other application state file to the user, allowing the user to delay the instantiation of the at least one other application state files, and loading the at least one other application state file, thereby allowing the user to manipulate a first aspect of the design before loading changes made to a second aspect of the design by another user.

25. (Original) The computer program product of claim 24 further comprising code for controlling instantiation of said received application state file into said local application.

26. (Currently amended) A computerized method for collaborating over a network to manipulate a design representing electrical or mechanical assemblies using a plurality of heterogeneous user applications running on respective clients connected to the network, said method comprising the steps of:

connecting a session client process to a session manager over the network to participate in a collaborative session;

sharing session control messages with other session client processes connected to said session manager;

loading design data representing said design into a local application running on said client;

creating at least one local application state event representing at least one application state of said local application based on at least one manipulation of said design using said local application; and

communicating said at least one local application state event from said session client process to said other session client processes via said session manager;

receiving at least one other application state file from said other session client, presenting the at least one other application state file to a user, allowing the user to delay the instantiation of the at least one other application state file, and loading the at least one other application state file, thereby allowing the user to manipulate a first aspect of the design before loading changes made to a second aspect of the design by another user.

27. (Previously presented) The method of claim 26 wherein the step of communicating further comprises:
communicating said at least one local application state event as at least one data packet for representing said at least one application state event.
28. (Previously presented) The method of claim 26 wherein the at least one local application state event is at least one of a plurality of normalized application state events recognized by each of the heterogeneous user applications.
29. (Previously presented) The method of claim 26 further comprising the step of:
saving said session control messages and said at least one local application state event in a journal file.
30. (New) The method of claim 1, further comprising the step of:
allowing the user to refuse the instantiation of the at least one application state file created by other local applications.
31. (New) The method of claim 1, further comprising the step of:
buffering the at least one application state file created by other local applications.